

A Distributed IoT Infrastructure to Test and Deploy Real-Time Demand Response in Smart Grids

*Original*

A Distributed IoT Infrastructure to Test and Deploy Real-Time Demand Response in Smart Grids / Barbierato, L., Esteban, A., Pons, E., Pau, M., Salassa, F.G.M., Ghirardi, M., Patti, E.. - In: IEEE INTERNET OF THINGS JOURNAL. - ISSN 2327-4662. - STAMPA. - 6:1(2019), pp. 1136-1146. [10.1109/JIOT.2018.2867511]

*Availability:*

This version is available at: 11583/2712099 since: 2020-01-17T15:04:08Z

*Publisher:*

IEEE

*Published*

DOI:10.1109/JIOT.2018.2867511

*Terms of use:*

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

*Publisher copyright*

IEEE postprint/Author's Accepted Manuscript

©2019 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collecting works, for resale or lists, or reuse of any copyrighted component of this work in other works.

(Article begins on next page)

# A Distributed IoT Infrastructure to Test and Deploy Real-Time Demand Response in Smart Grids

Luca Barbierato, Abouzar Estebarsari, Enrico Pons, Marco Pau, Fabio Salassa, Marco Ghirardi and Edoardo Patti

**Abstract**—In this paper, we present a novel distributed framework for real time management and co-simulation of Demand Response (DR) in smart grids. Our solution provides a (near-) real-time co-simulation platform to validate new DR-policies exploiting Internet-of-Things approach performing software-in-the-loop. Hence, the behavior of real-world power systems can be emulated in a very realistic way and different DR-policies can be easily deployed and/or replaced in a plug-and-play fashion, without affecting the rest of the framework. In addition, our solution integrates real internet-connected smart devices deployed at customer premises and along the Smart Grid to retrieve energy information and send actuation commands. Thus, the framework is also ready to manage DR in a real-world Smart Grid. This is demonstrated on a realistic smart grid with a test case DR-policy.

**Index Terms**—Internet-of-Things, Smart Grid, Demand Response, co-simulation, real-time simulation, distributed infrastructure, Smart Metering Architecture

## I. INTRODUCTION

Current European electrical grids are experiencing several issues due to the rise of electricity demand, that is growing continuously with a predicted remarkable rate of 1.1% per year [1]. Renewable Energy Sources (RES) integration and diffusion increase the uncertainty in electricity dispatch plan and control due to the intrinsic fluctuation in the electricity generation pattern of these resources. To cope with these issues, different solutions have been proposed that ensure reliability of the grid management, such as physical storage [2] (e.g. batteries) and virtual power plants [3].

Even if physical storage is assessed to be of great value for grid management, the cost of large storage systems is too high for a massive deployment in electrical grids. On the other hand, virtual power plants are considered a cost-effective and feasible way to solve issues on grid management. Demand Response (DR) [4] permits achieving a temporary virtual power plant by changing the energy consumption pattern of consumers to fulfil grid operation requirements or economical incentives. In literature, DR policies are classified depending on possible objectives [5], [6] (e.g. economy driven,

reliability driven and ancillary services driven) or based on control strategy [7], [8], [9] (e.g. based on prices, demand-dispatch, direct and autonomous control). End-user flexibility is often exploited through day ahead or intra-day scheduling to balance the network, but (near-) real-time DR programs may also exist to deal with unexpected contingencies or loss of reliability in the network. Most of the current DR plans refer to large commercial and industrial customers [10] that sell their electrical flexibility in energy markets. However, residential DR is becoming an important key point in grid management, as electricity demand of households will reach 23% of the total electricity demand in the next decade [1]. DR programs also involving residential customers are already active in US [11], while in Europe they are not yet deployed, mainly due to the lack of a suitable regulatory framework. However, it is well recognized that DR can be a strategic tool to improve energy efficiency and to have optimal use of the grid assets [12], [13]. DR can be thus expected to gain importance also in Europe in the next future and to play a key role for addressing the challenges brought by the evolving scenario of the distribution grids.

To provide a better management of distribution networks and RES, future Smart Grids will be equipped with pervasive Internet connected devices following the rising Internet-of-Things (IoT) approach. In the IoT vision, *anything can be connected anywhere into the global information network* (i.e. the Internet) *at anytime* [14]. Thus, IoT will be a preferred medium to transmit sensed information and actuation commands [15] enabling a two-way communication among Smart Grids devices and components [16], [17], [18]. This combination of Smart Grid view together with distributed RES and IoT technologies is also known as *Internet-of-Energy* [19], [8]. These infrastructures have to be designed to deal with Big Data transmission and processing [15]. Furthermore, software components of these infrastructures can be optionally deployed in cloud systems [20]. Cloud computing offers different solutions to access shared pools of configurable ICT resources (e.g. servers, storages and applications), which can be easily and quickly provisioned with minimal management effort.

In this scenario, Advanced Metering Infrastructure (AMI) [21] and IoT devices (i.e. smart meters [22] and smart appliances), are key technologies to foster novel services in Smart Grid [23], such as the DR in residential contexts. On these regards, smart devices are part of AMI and allow a fine grained collection of energy measurements. In particular, novel smart meters [22] can sample data spanning in the range of  $1s$  to  $15min$  [24], depending on the services to provide. Disaggregating and post-processing these measurements

L. Barbierato and E. Patti are with the Department of Control and Computer Engineering, Politecnico di Torino, Torino, Italy.

A. Estebarsari and E. Pons are with the Department of Energy, Politecnico di Torino, Torino, Italy.

F. Salassa and M. Ghirardi are with the Department of Management and Production Engineering, Politecnico di Torino, Torino, Italy.

M. Pau is with the Institute for Automation of Complex Power Systems, RWTH Aachen University, Aachen, Germany

Emails: {luca.barbierato, abouzar.estebarsari, enrico.pons, fabio.salassa, marco.ghirardi, edoardo.patti}@polito.it, {mpau}@eonerc.rwth-aachen.de

This work was supported by FLEXMETER, which is an EU Horizon 2020 project under grant agreement no. 646568.

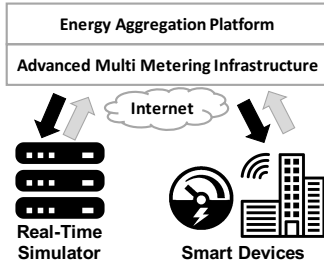


Fig. 1. Scheme of the proposed framework

allow retrieving information about consumption behavior in households, such as appliance activation and energy usage [25]. These post-processed data can feed other novel energy services, such as DR in households [26].

In this paper, we present a novel framework (see Fig. 1) which can serve DR and Demand Side Management (DSM) functions in smart grids from two aspects: i) providing a (near-) real-time co-simulation platform for validation of new algorithms through a so-called real-time software-in-the-loop set-up which emulates the behavior of the real-world system very realistically; ii) integrating real IoT devices deployed at customer premises to retrieve and collect energy information. This framework integrates our proposed cloud-based *Advanced Multi-metering Infrastructure* (a.k.a. FLEXMETER) with one or more Digital Real-Time Simulators (DRTS) which accommodate grid models. Considering the capability of parallel computation in Real-Time Simulation (RTS) through clustering several DRTS, and according to the new experiments on geographically distributed real time co-simulations ([27], [28], [29]) in which distant laboratories interconnect their DRTS to enhance calculation power of the integrated simulation platform, large electricity grids with a lot of elements can be modeled and simulated for ex-ante laboratory tests. This capability allows replacing DRTSs with the real-world Smart Grid environment where the proposed framework is intended to be used too. It provides a bidirectional (near-) real-time communication with real IoT devices, and enables the interoperability among heterogeneous technologies. Regarding the simulation aspect of our framework, the novelty can be summarized as: i) different DR algorithms can be easily assessed in a very realistic testbed in a plug-and-play fashion; ii) interoperability of DR algorithms with other smart grid control and management strategies can be evaluated without integrating all modules in a single monolithic software or program; iii) integrating DRTS provides a very accurate and efficient simulation of smart grid; iv) different modules, hosting control algorithms or time-variant parameters of smart grids, connected to this framework can be integrated while running on different Software/Hardware platforms; v) IoT based approach used in the framework architecture enables clustering several DRTSs to enhance overall computation power and provide concurrent simulation of different levels/subsystems of a smart grid, e.g. Medium Voltage (MV) grids, Low Voltage (LV) grids and residential households. Moreover, exploiting FLEXMETER infrastructure, the simulated scenario in RTS can

be fed with real energy information coming from real IoT devices installed across the distribution network providing a real-time management of DR in smart grids. The rest of this paper is organized as follows. Section II reviews most relevant literature solutions. Section III describes in-depth each module of our proposed framework. Section IV introduces the DR algorithm that we used as a test-case. Section V presents both case study and experimental results. Finally, Section VI discusses concluding remarks.

## II. RELATED WORK

New generation smart meters are key enabler of AMI and foster new energy related services such as DR. In [30], [31], authors present two Smart Meter Systems that allow a bidirectional communication with a centralized DR management platform. Aguirre et al. [30] present a new generation smart meter designed to support new requirements for operation and control of the distribution network grid. LeMay et al. [31] describe a Meter Gateway Architecture for integrated control of loads by energy aggregators. Not only Smart Meters are important in such context. For instance, Mashima et al. [32] present a DR System Framework leveraging on a DR client mobile app able to directly control IoT devices according to user policies.

Different DR Infrastructures have been proposed to cope with the necessity of solutions for Smart Grid management and control. In [33], Kim et al. introduce a cloud-based DR platform able to perform power reduction requests and to return the price incentive per customer to obtain it. Jacobsen et al. [34] present SEMIAH, a scalable infrastructure for residential DR exploiting shiftability of smart appliances within a Home Energy Management Gateway in communication with a distributed platform that permits receiving commands from a DR system to postpone appliance activation. In [35], Bhattarai et al. designed a hierarchical DR architecture. This infrastructure is able to perform three levels of control and actuation to manage residential DR resources (e.g. Plug-in Hybrid Electric Vehicle) using in each level a different kind of DR techniques (i.e. Direct Load Control, Price based, Demand Dispatch and Autonomous).

Müller et al. [36] highlighted the important role of the Information and Communication Technology (ICT) in simulating future power systems. Indeed, Smart Grids are complex systems where different entities cooperate by exchanging heterogeneous information. In this view, all the previous solutions need to be assessed in a simulated environment to confirm DR service feasibility in terms of network communication, data management and result on Smart Grid behaviour. Yang et al. [37] propose a co-simulation environment to validate distributed controls in Smart Grid. To build closed-loop models, they develop controllers in MATLAB/Simulink that communicate with power plants models through either UDP or TCP protocols. Manbachi et al. [38] present a co-simulation platform for evaluating the performance of a volt-VAR optimization engine. In its core, it integrates a real-time simulator for distribution networks with physical devices for Measurement & Control. MOSAIK [39], [40], [41] is

TABLE I  
COMPARISON AMONG OUR CO-SIMULATION INFRASTRUCTURE AND  
LITERATURE SOLUTIONS

Solution	Environment for DR			AMI Integration		RTS	IoT
	simulation	real-world Smart Grid	DER	Data collection	Actuation		
Our framework	✓	✓	✓	✓	✓	✓	✓
Aguirre et al. [30]		✓		✓	✓		✓
LeMay et al. [31]							✓
Mashima et al. [32]		✓			✓		
Kim et al. [33]		✓					
Jacobsen et al. [34]		✓			✓		
Bhattarai et al. [35]	✓	✓			✓	✓	
MOSAIK [39], [40], [41]	✓		✓				
GRIDspice [42]	✓		✓				
ENEL [43]	✓					✓	
Faria et al. [44]	✓		✓				
Tan et al. [45]	✓		✓				
Wijaya et al. [46]	✓						

a flexible architecture to manage control strategies, scenario specifications and simulation models by exploiting semantic knowledge. To analyse power-flows in smart grids, MOSAIK integrates a co-simulation between the software *PowerFactory* and their model (in Matlab) for both Photovoltaic and Load generations. GRIDspice [42] is a distributed platform that combines existing simulation software for power generation and distribution (i.e. MATPOWER and Gridlab-D). It exploits a cloud-based architecture that allows the parallelization of large simulations. Both MOSAIK and GRIDspice offer a flexible and scalable simulation framework but they do not provide real-time features for short-transient phenomena. Finally, in the context of smart grids real-time simulations, ENEL (the biggest Italian utility) is running the POI P3 Smart Grid project [43] to test and validate future smart grids. In particular, ENEL is testing new techniques for voltage regulation through SCADA and Distribution Management System. Faria et al. [44] present DemSi, a novel DR simulator allowing studying different DR actions. Tan et al. [45] introduce Smart Grid Common Open Research Emulator (SCORE). SCORE permits emulating power and communication network in the Smart Grid context to bridge the gap between testbed and simulation. In addition, both Faria et al. [44] and Tan et al. [45] include in their solutions Distributed Energy Resources (DER). Another solution centered on DR assessment is DRSim [46], a cyber-physical simulator for DR systems. DRSim is able to model the emergent behavior of a community based on real data traces, such as household consumption patterns, appliance usage behaviors, IoT sensor data (e.g. motion sensor) and price signals.

These presented solutions are not capable to perform simulation of DR policies integrating also RTS in their frameworks. They also lack in taking advantages from the integration of real data coming from real IoT devices (i.e. smart meters and smart appliances) and/or AMI. Indeed, they should integrate IoT communication paradigms and protocols to perform both Hardware- and Software-in-the-Loop co-simulations to be fully compliant with the *Internet-of-Energy* view.

Considering three DR related trends in the context of smart grids, namely smart meter utilization for DR, deployment of new DR and DSM infrastructures, and the necessity of validation of new DR algorithm through reliable simulations, we propose a generic solution to address all these trends: our proposed real-time co-simulation framework is able to

simulate different algorithms through the *Energy Aggregation Platform* (EAP) that is a "virtual box" where different DR-policies are executed. EAP eases the replacement of a DR-policy with another without affecting the rest of the platform. This is a novel simulation approach whose importance is stressed in case DR developer has no interest or expertise in grid modelling or other involved smart grid actors. The other highlight of this co-simulation platform, thanks to its IoT-based approach, is its flexibility in integrating many different simulation modules and modelling tools, even several real-time simulators as a cluster of grid simulators. The latter feature has a great added value in simulating a large multi-level system (e.g. simulation of a system formed by an MV grid, several LV grids, residential building units, individual households, commercial and industrial consumers, etc.). Integrating DRTS brings such capabilities of distributed co-simulations, either local or geographically distant, (e.g. [28], [29]), and also of remote testing of algorithms (e.g. [27]). This framework is also ready to be used in a real-world Smart Grid environment either by replacing RTS with real grid (i.e. hardware and/or software components), or replacing virtual measurements in the model with real data from smart meters. Indeed, the proposed framework integrates also FLEXMETER, which is our Advanced Metering Infrastructure. FLEXMETER already integrates different technologies and protocols used by novel commercial smart meters and smart appliances. The integration of DRTS actually allows simultaneous simulations of realistic Smart Grid scenarios in parallel with real data acquisition from smart meters (load energy consumption or generation energy production from renewable systems) to assess novel energy services and evaluate new control strategies with much higher accuracy than any other pure laboratory/simulation tests.

To highlight our contribution, Table I reports a comparison of our framework with reviewed literature solutions for DR co-simulation. It highlights: i) if the environment to run DR-policies consists on simulation tools or they can be applied also in a real world smart grid or they integrate DER; ii) if AMI is integrated and provides features to collect data from smart devices and send actuation commands; iii) if a Real-time simulator is used; and iv) if the infrastructure interacts with IoT devices (i.e. smart meters and smart appliances).

### III. DISTRIBUTED FRAMEWORK FOR DEMAND RESPONSE

In this section, we describe the components of the proposed distributed framework. As shown in Fig. 1, we identify three main components: i) *Advanced Multi-metering Infrastructure*, ii) *Energy Aggregation Platform* and iii) *Real-Time Simulator*.

#### A. Advanced Multi-metering Infrastructure

The *Energy Aggregation Platform* leverages upon our *Advanced Multi-metering Infrastructure*, called FLEXMETER, that fosters general purpose services in a smart grid scenario. We presented FLEXMETER in our previous work [47], where we described a solution for distributed state estimations that works together with a service for automatic network reconfiguration. Whilst, in this paper we describe a novel solution for (near-) real-time management and co-simulation of

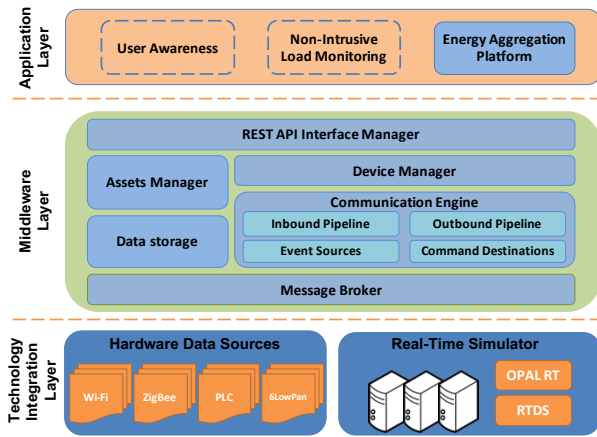


Fig. 2. Scheme of FLEXMETER infrastructure

DR-policies in smart grids. Thus, Demand Response through *Energy Aggregation Platform* is another service leveraging upon FLEXMETER that is briefly introduced in this section to give a complete overview of the proposed solution for co-simulation of DR algorithms.

FLEXMETER’s architecture has been designed to cope with interoperability among heterogeneous smart devices abstracting different underlying low-level technologies. This enables a fine-grain monitoring of the overall energy distribution network, offering a common data access to this large amount of incoming information. This provides *Energy Aggregation Platform* with necessary information (i.e. appliance start-time and power profile of buildings or substations) to perform DR services. FLEXMETER has been designed as a single instance that can optionally run in a cluster of servers or in a cloud system allowing multi-tenancy [48].

As shown in Fig. 2, FLEXMETER is a three-layered architecture with: i) a *Technology Integration Layer*, ii) a *Middleware Layer* and iii) an *Application Layer*.

1) The **Technology Integration Layer** is the lower layer of FLEXMETER. It is made up of different *Technology Integration Adapters* (TIA), one for each technology, developed following a methodology proposed in [49]. TIA is a middleware-based software component in charge of integrating heterogeneous hardware data sources that exploit different communication protocols (e.g. IEEE 802.11, ZigBee or 6LowPan). It abstracts device features and functionalities providing common interfaces to access them. In addition, TIAs harmonize information coming from low-level devices in a common data-format before sending it to the rest of the platform via MQTT [50]. MQTT stands for Message Queuing Telemetry Transport. It is a communication protocol that implements the publish/subscribe paradigm [51]. We developed also specific TIAs to integrate DRTSSs, such as RTDS and Opal-RT. Virtual devices simulated on DRTSs are transparently seen by the other FLEXMETER’s actors as IoT devices ready to communicate over the Internet. Thus for the infrastructure, there is no difference in dealing with a real or a virtual IoT devices. This allows i) an easy switch from a simulation environment to a real-world smart grid, and vice-versa, and ii) a bidirectional

data exchange between real-world devices and RTS, even in (near-) real-time.

2) The **Middleware Layer** in Fig. 2 consists of several software components acting together to: i) ensure bidirectional communication with TIAs; ii) collect, check and store data messages; iii) provide unified interfaces to access data, devices, assets and maintenance operations and iv) send commands to devices. The *Message Broker* manages the MQTT communication with devices allowing asynchronous bidirectional communication with the cloud infrastructure. It routes all messages to the *Communication Engine* that manages the communication between the platform and the devices. In its core, *Event Sources* receives measurements and alerts from TIAs, thus from devices, and checks their integrity before pushing them into the *Inbound Pipeline* that prevents congestions in storing information into the *Data Storage*. The *Outbound Pipeline* manages out-coming messages to be sent to devices through the *Command Destinations*. The *Data Storage* is a module that integrates different non-relational database models to store information. The *Device Management* supervises the device provisioning handled by the application layer. Whilst, the *Asset Modules* handle information of the assets in FLEXMETER (e.g. people, places and things). Finally, applications retrieve data and manage entities or devices from the platform through the *REST API Interface Manager* that provides REST (Representational State Transfer) web services [52].

3) The **Application Layer** offers APIs and tools to develop services and applications to manage and post-process information coming from the underlying layers. For example, *User Awareness service* allows end-users in retrieving information about their energy behaviour patterns [53]. *Non-Intrusive Load Monitoring*[25] disaggregates home appliances load consumption from a single point of measurement given by the smart meter. *Energy Aggregation Platform* is a complex service to perform DR in Smart Grid (see Section III-B). This layer promotes also the data exchange among services.

### B. Energy Aggregation Platform

The *Energy Aggregation Platform* (EAP) is a software designed to provide novel services for smart grid management (i.e. DR and DSM) to different stakeholders playing in the energy marketplace (e.g. Distribution System Operators, Retailers and Energy Aggregators). EAP is a “virtual box” that gives to the whole framework the flexibility to deploy or replace easily a DR-policy without affecting the rest of the framework. EAP exploits the FLEXMETER infrastructure that provides APIs and tools to abstract functionality of devices, either real or virtual, and transparently manage information, as pointed out in Section III-A. In addition, EAP provides common software interfaces to establish a bidirectional communication among DR-policies and smart devices. Thus, DR-policies easily retrieve energy information and send actuation commands to smart devices in (near-) real-time neglecting their low-level technologies (i.e. hardware and protocols). EAP is developed following the multi-tenant design pattern approach [48]. As shown in Fig. 3, it consists of five modules described in the following.

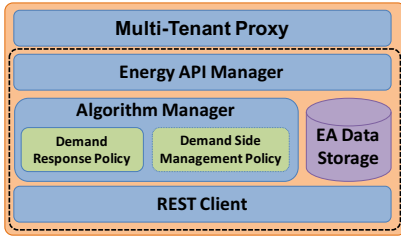


Fig. 3. Scheme of Energy Aggregation Platform

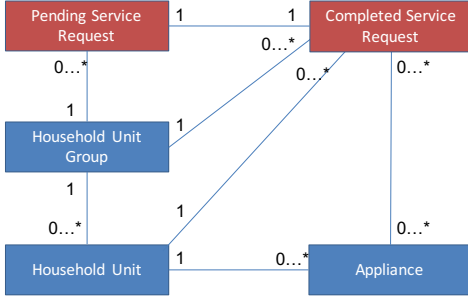


Fig. 4. Energy Aggregation Platform Data Model

The *REST Client* retrieves information on assets and measurements from FLEXMETER’s *REST API Interface Manager*. Energy measurements are collected in (near-) real-time by FLEXMETER that receives them from smart devices through MQTT protocol. The *REST Client* is also used i) to populate or update information in the *EA Data Storage* module and ii) to send actuation commands to smart devices, again through FLEXMETER.

The *EA Data Storage* module exploits a non-relational database (e.g. MongoDB) and stores information needed by DR-policies following the data model in Fig. 4. This data model describes customers, called *Household Units*, involved in the DR strategy. Each instance of *Household Unit* is in relation with a set of *Appliances* that might participate to a DR-policy. *Household Units* are grouped in *Households Unit Groups* depending on the relations between customers (e.g. customers belonging to a common substation or to a common retailer). Each *Household Unit Group* is in relation with i) a set of *Pending Service Request* and ii) a set of *Completed Service Request*. The *Pending Service Request* represents an action to be performed in a *Household Unit Group* (i.e. DR-event) triggered by stakeholders. Once the action is performed by DR-policy, the *Pending Service Request* becomes a *Completed Service Request* that includes additional information on the results of the fulfilled event.

The *Energy API* module offers REST APIs to create a *Pending Service Request* and to retrieve all the information in the *EA Data Storage*.

The *Algorithm Manager* is the “virtual box” that contains and executes policies for both DR and DSM events. In order to perform the DR-policies’ routines, this module defines and provides software interfaces to retrieve and update all the information in the *EA Data Storage* (e.g. *Household Units* involved in a specific DR-event). Once new actions must be performed, the *Algorithm Manager* posts the actuation com-

mands (result of DR-policies) to FLEXMETER by exploiting the *REST Client* module. Then, FLEXMETER will send these commands to the right smart devices via MQTT protocol. This process is performed in (near-) real-time. Thanks to this module, new DR-policies can be easily deployed or replaced without affecting the rest of the framework.

Finally, the *Multi-Tenant Proxy* manages stakeholders authentications before creating a new instance of EAP (represented by the dashed-line macro-block in Fig. 3). Each EAP tenant handles its own set of *Household Unit Groups* with a specific DR-algorithm. Thus, different EAP tenants are executed at the same time and each of them manages different stakeholders’ customers avoiding data intermingling.

### C. Smart grid real time simulator

The proposed distributed framework is ready to work in a real Smart Grid environment. Thus, it is able to retrieve energy data from real smart devices and send them back actuation commands that are outputs of the DR-policy running on EAP. To evaluate novel DR-policies and their impacts on the Smart Grid, this framework offers a simulation engine consisting of one or several DRTS, such as RTDS or OPAL-RT. In this scenario, FLEXMETER infrastructure provides common interfaces to enable a bidirectional communication with smart devices and collects energy information. RTS realistically reproduces the behavior of Smart Grids with high accuracy and efficiency. This is gained due to three main reasons: parallel and distributed simulation capability, different multicore processors (FPGA in OPAL-RT, or NovaCor in RTDS) with respect to normal CPUs (in PCs), and advanced and optimal solvers to perform discrete-time simulation with very narrow time-steps (in range of micro- to milli- seconds) efficiently. Using our IoT-based framework a cluster of DRTS can concurrently run being located either in the same lab or in several laboratories even geographically distant. This not only increases computation power which is highly demanded by large-scale grids with many components and actors, but also enables concurrent simulation of different hierarchical levels of smart grids by running on several simulators (e.g. MV grids, LV grids, building units, households, etc.). In our framework, hard real-time simulations are performed by DRTS, where the power system model is running in time domain, while the rest of the modules involved in the co-simulation are running in (near-) real-time. Since the communication delay in the co-simulation infrastructure is often much smaller than modules’ responses/interactions, synchronization can be ensured if real-time simulators are running with a shifted time stamp. Time-variant parameters (e.g. energy consumption, generation output, control commands, etc.) can be periodically pushed into the running real-time model to observe Smart Grid operational behaviour. However, such interactions are not necessarily done with the same rate of simulation sampling: for Hardware-In-the-Loop (HIL) a higher frequency of sampling is needed, while for Distribution Management System applications the frequency could be lower. In our framework, we use RTS with its narrow time-steps for mainly two reasons: enabling the platform to perform HIL tests, especially

with embedded system controllers or smart meters, and slow transient analysis of the electricity grid in terms of voltage or frequency deviation by interoperability studies of demand response algorithms and other grid control and management strategies (e.g. Volt-VAR control). It can also replicate the energy consumption patterns in households exploiting the real energy information coming in (near-) real-time from smart devices through FLEXMETER.

As mentioned in Section III-A, RTS communicates with the rest of the framework by exploiting a specific TIA developed filling a methodology proposed in our previous work [54]. This TIA implements the two communication paradigms provided by FLEXMETER: i) *publish/subscribe* [51] based on MQTT protocol [50] and ii) *request/response* based on REST [52]. Finally, once the DR-policy performs some actions, the EAP translates these actions into actuation commands that are sent to the virtual smart devices that are simulated in RTS, through the FLEXMETER infrastructure.

For the preliminary testing and validation of the DR policies, as an alternative to real smart devices, virtual smart meters and virtual smart appliances should be used in the RTS environment. Therefore we developed a profile generator able to create realistic patterns of consumption for residential customers, down to the single appliances, allowing the emulation of the real behavior of households in the smart grid. The profile generator can create multiple days of power consumption data, thus enabling long term simulations, and uses real consumption data of single appliances to generate the overall power consumption profile. Generated data provide the time-variant set points of active and reactive power for each household, which are used by the DRTS engine for the simulation of the grid operating conditions.

A dedicated software module acts as controller of the simulation, handling the proper transmission of the household consumption set points to the RTS. The same software module also integrates a TIA and emulates the virtual smart devices targeted by the DR commands sent by the EAP.

Once a DR-event is triggered, the simulation controller module receives via MQTT, through FLEXMETER, the actuation commands issued by the EAP. It then processes the command (e.g. shut down or reactivation of an appliance) accordingly modifying the consumption profile of the appliance, and updates the set points of the associated household, which are subsequently forwarded to RTS.

#### D. Deployment considerations

As pointed out in the previous sections, the proposed framework for Demand Response follows the modern software design patterns (e.g. multi-tenancy [48]) to build distributed architecture that can be deployed on cloud systems.

In this view, FLEXMETER infrastructure can be deployed in cloud systems and offered to stakeholders as *Platform as a Service* (PaaS). PaaS is a category of cloud computing services that provides all the facilities required to support the complete application life-cycle. Thus, different stakeholders (e.g. Distributor System Operators, Energy Aggregators and Retailers) develop, run and manage their applications and services without the complexity of building and maintaining the

whole FLEXMETER infrastructure. As an alternative, PaaS can also be delivered providing all the software to be installed in private data-centers managed by internal IT departments [20]. This second deployment approach could minimize the data exchange over the Internet among components (either hardware or software) and FLEXMETER infrastructure.

EAP can be deployed in cloud systems and offered to stakeholders as *Software as a Service* (SaaS). SaaS is a software delivery methodology that provides virtualized computing resources over the Internet. SaaS architecture provides a complete software solution, including set-up and infrastructure management. SaaS is a software distribution model where a producer develops, operates and manages an application, making it available to its customers via the Internet. Thus, EAP can be given with a set of predefined DR-policies that can be easily customized, updated or replaced by stakeholders according to their requirements and use cases. As an alternative, EAP can also be delivered as software to be deployed in private data-centers managed by stakeholder's IT departments [20].

It is worth noting that the different hardware and software components can be installed in various locations geographically closed or distant, and they communicate over the Internet following the *Internet-of-Energy* view. Thus, FLEXMETER platform, the different EAP instances and the real-time simulators can be equally installed and deployed in the same data center or in different cloud systems. These deployment choices will not affect the behaviour of the overall proposed framework.

#### IV. GREEDY ALGORITHM FOR DEMAND RESPONSE

As mentioned in Section III-B, EAP is a "virtual box" to easily deploy and/or replace different DR-policies. To better characterize the functionalities of EAP, we designed and run experiments with an effective DR procedure belonging to the paradigm of Greedy Algorithms [55], hereinafter referred to as GrAI. Greedy Algorithms are solution approaches relying on local optimal choices with the hope of finding a global optimum. Main advantages are the easy design and the limited computational running times. We point out that GrAI is just one of the possible algorithms that can be embedded into EAP "virtual box" and ran in our framework for DR. More complex algorithms can be developed and executed following, for instance, the methodologies presented in [5], [7], [8].

The solution approach is based on a two-stage procedure activated whenever a DR-event for overall exceeding energy consumption is identified. In both stages of the procedure the problem is modeled as a Knapsack Problem with binary variables (KP0/1) [56], and a classical greedy approach is then applied two times to the modeled KP0/1 problem. Once a DR-event happens, the GrAI reads all active appliances starting times in every *Household Unit* considered in the specific scenario and computes the overall energy consumption from the DR-event time-stamp on to a desired time horizon (e.g. 15, 30 or 60 minutes). As an input, GrAI also takes a desirable (feasible) cumulative energy consumption, thus it is able to compute the total amount of energy that should be cut-off via appliances shut down. Then, for each *Household Unit*,

GrAI sums the relative consumption of the appliances so as to have the global consumption for each house. The algorithm then orders *Household Units* considering both the global consumption and the relative costs, represented by a single discomfort integer parameter when selecting a specific house, and selects the ones providing the least global discomfort for end users with a global energy save greater than the requested cut-off. The *discomfort parameter* for *Household Units* has been introduced to make possible not to select every time the same set of them. This can be achieved by modifying the value of the parameter after a DR-event, leading to a different ordering of *Household Units* during GrAI runs.

In the second stage, among the selected *Household Units*, the algorithm selects the minimum set of appliances ordering them with respect to a discomfort parameter for end-users providing the least number of appliances that guarantee that the global cut-off is respected. Also in this case, the *discomfort parameter* for *end-users* is introduced to weight differently the cost of the appliances involved in the shut-down if necessary. The reactivation phase is performed as the inverse priority order of appliances shut down, providing the most important appliances to be switched on first.

The detailed GrAI procedure is presented in Algorithm 1. The main steps are the following:

- Compute the average consumption of all appliances activated during the considered event (steps 2-4).
- Solve a KP0/1 problem on buildings, to select the involved buildings (steps 5-14)
- Solve a KP0/1 problem on all the appliances of the selected buildings, to select the appliances to be switched off (steps 15-24)
- Give a timing of switching off (in non-decreasing discomfort order - steps 25-33), and on again (in non-increasing discomfort order, steps 34-39). In order to obtain a smooth power cut in time, the time passing after one appliance switched off to the next is proportional to the amount of energy cut. Analogously, the time passing after an appliance is switched on to the next is proportional to the amount of energy added.

## V. EXPERIMENTAL RESULTS

Test results presented in the following are generated through emulation of an LV smart grid subtending 360 residential customers in RTS environment. The considered grid is a portion of a real distribution network in a semi-rural area in north of Italy, and it is composed of 20 nodes out of which 15 nodes are connected to residential customers (Fig. 5). The total number of households is approximately equally distributed among the nodes, so that each node subtends between 20 and 28 customers. The size and the ratio of the MV/LV transformer in the upstream secondary substation is 250 kVA and 22/0.4 kV respectively. Resistance and reactance (R, X) of lines with cross-sections 16, 50, and 95 *mmq* are (1.16, 0.0817), (0.391, 0.0779), and (0.19, 0.0751) ohm/km respectively.

For this scenario, daily power consumption profiles of the households are created using the load profile generator mentioned in Section III-C. For each customer, a set of appliances existing within the house has been extracted taking into

### Algorithm 1 GrAI

```

1: INPUT:  $a_{i,j}$ : set of appliances ( $j$ -th appliance out of  $n_i$  appliances of building  $i = 1..N$ );  $f_i$ : global discomfort of switching off building  $i$ ;  $p_{ij}$ : discomfort of switching off the  $j$ -th appliance (out of  $n_i$ ) of building  $i$ ;  $b$ : maximum consumption level after the switching off.  $T_{DR}^{Event}$  and  $T_{ON}$ : time for beginning the switch-off/switch-on.  $\Delta_T^{OFF}$  and  $\Delta_T^{OFF}$ : time available for the switch-off/switch-on;  $c_{ij,t}$ : consumption of appliance  $a_{i,j}$  at time  $t$ .
2: for  $i = 1..N, j = 1..n_i$  do
3:    $w_{ij} = \frac{1}{\Delta_T^{OFF}} \sum_{t=T_{DR}^{Event}}^{T_{DR}^{Event} + \Delta_T^{OFF}} c_{i,j,t}$ 
4: end for
5:  $A = \emptyset$  (Initialization of the switching off candidate set)
6: Order buildings  $i$  in non increasing order of  $\frac{f_i}{\sum_{j=0}^{n_i} w_{ij}}$ 
7:  $currentload = 0$ 
8: for  $i = 1..N$  do
9:   if  $currentload + \sum_{j=0}^{n_i} w_{ij} < b$  then
10:     $currentload = currentload + \sum_{j=0}^{n_i} w_{ij}$ 
11:   else
12:    Add to set  $A$  all appliances belonging to building  $i$ .
13:   end if
14: end for
15: Order  $a_{i,j} \in A$  in non increasing order of  $\frac{p_{ij}}{w_{ij}}$ 
16:  $currentload = 0$ 
17: for  $i = 1, \dots, size(A)$  do
18:   if  $currentload + w_{ij} < b$  then
19:     $currentload = currentload + w_{ij}$ 
20:     $x_{ij} = 1$  (appliance not to be switched off)
21:   else
22:     $x_{ij} = 0$  (selected for switching off)
23:   end if
24: end for
25: Set  $A = \{a_{i,j} : x_{ij} = 0\}$  (appliances to be switched off).
26: Set  $w_{tot} = \sum_{(i,j) \in A} w_{ij}$ .
27: Order  $a_{i,j} \in A$  in non decreasing order of  $p_{i,j}$ .
28:  $T_{curr} = T_{DR}^{Event}$ 
29: for  $k = 1, \dots, size(A)$  do
30:   Let  $(i, j)$  be the indexes of appliance  $A_k$ .
31:   Set the switch off for appliance  $a_{i,j}$  at time  $T_{curr}$ .
32:    $T_{curr} = T_{curr} + \frac{\Delta_T^{OFF}}{w_{tot}} w_{ij}$ 
33: end for
34:  $T_{curr} = T_{ON}$ 
35: for  $k = size(A), \dots, 1$  do
36:   Let  $(i, j)$  be the indexes of appliance  $A_k$ .
37:    $T_{curr} = T_{curr} + \frac{\Delta_T^{ON}}{w_{tot}} w_{ij}$ 
38:   Set the switch on for appliance  $a_{i,j}$  at time  $T_{curr}$ .
39: end for
40: OUTPUT: Off/On Timings set for selected appliances.
    
```

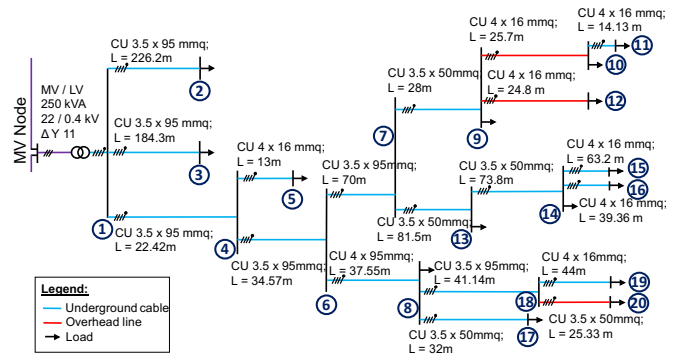


Fig. 5. Topology and line parameters of the grid.

account user-defined percentages of diffusion of the appliance. As an example, 100% of the customers has been assumed to own a fridge, whereas only 50% of the customers has been considered to have a dishwasher. The overall power

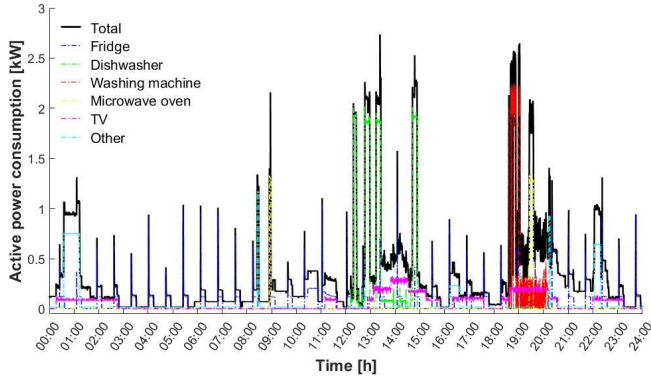


Fig. 6. Example of power consumption profile generation for a household.

consumption profile for each household is obtained as the aggregation of the single appliances' consumption, which include a large set of devices such as fridge, washing machine, dishwasher, microwave oven, TV, etc. In particular, in the performed simulations, the following appliances are considered as smart IoT devices shiftable through DR policies: fridge, stand-alone freezer, washing machine, tumble dryer, dishwasher and electric water heater. Fig. 6 shows, as an example, the consumption profile generated for one of the households, where the contribution of the main energy-hungry devices, like washing machine and dishwasher, can be identified.

Given this set-up, DRTS can run the smart grid emulation. Measurement data from the households virtual smart meters are sent to FLEXMETER every minute thanks to the TIA embedded in the simulation controller. In addition, time-tagged switch-on events from the shiftable smart devices are also sent to the cloud to provide EAP with the relevant data needed for the proper application of the DR logic. The simulation controller also has a specific sub-module in listening mode, which allows the collection of the actuation commands from the EAP whenever a DR-event is triggered. In this way, the household consumption profiles can be updated and used to assess the effects of the DR policy under test on the smart grid behavior.

#### A. Results on smart grid behavior

To investigate the impacts of DR on the behavior of smart grids, two simulation scenarios, with and without DR, were executed, and the power flow at the transformer and the voltage profiles at different nodes were monitored and captured every 1 minute for a 24-hour period.

The results of the scenario without DR show that the apparent power flowing through transformer is reaching the transformer capacity (80 to 96 percent) after 18:00. At the same time, voltage profiles of some far nodes report a slight violation of voltage (to less than 90 percent) during the peak load event. In the other scenario we triggered DR after 18:30 to shave peak loads. Operation of the appliances selected by the EAP was stopped for 2 hours and, therefore, reactivation occurs after 20:30. This load reduction during the peak time along with a consequent loss reduction in the grid resulted in a decrease of apparent power flow at the substation (Fig. 7).

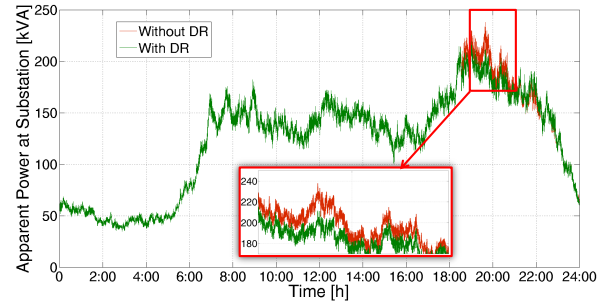


Fig. 7. Total apparent power at MV/LV substation with and without DR.

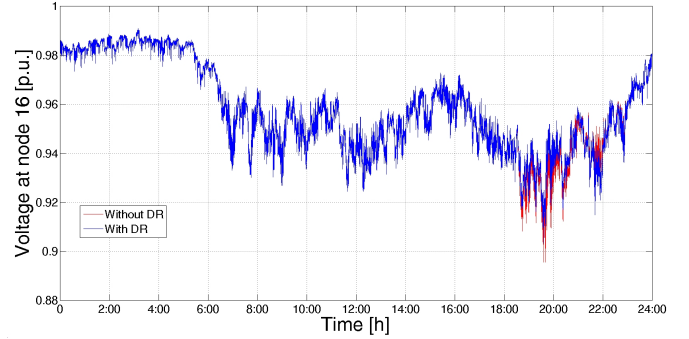


Fig. 8. Voltage profile at node 16.

Simultaneously, voltage profiles were also slightly improved (Fig. 8).

The small voltage drops in LV nodes might be negligible, and the risk of transformer congestion could be resolved by replacing it with a larger one. But considering trends in smart grids towards integrating more distributed energy resources (e.g. rooftop Photovoltaic panels in urban areas), total power demand of these LV systems from upstream networks would remain below transformers' capacity for longer time periods of a day. Therefore, it is wise to utilize smart devices and efficient DR algorithms to control the events instead of physically reinforcing costly infrastructure.

#### B. Performance analysis of GrAI

To estimate the maximum error generated by the heuristic algorithm of the DR-policy (see Section IV) w.r.t. to the precise percentage quantity. The test instances have been generated in the following way. The number of *Household Units* (i.e. customers) are 25, 50, 100 and 200 to represent different scales of the problem. The error is calculated as the ratio between the total consumption and the largest consuming appliance. The number of appliances is constant for each *Household Unit*, namely 6 different appliances. We present results up to 200 *Household Units* because these are primarily affected by error. The worst case is thus represented by the scenario where the appliance selected to exceed the cut-off is the most consuming one, giving the ratio as explained. We recall that this is a theoretical worst case analysis since appliances, before being selected, are ordered in decreasing order w.r.t consumption, thus the real error should be smaller.

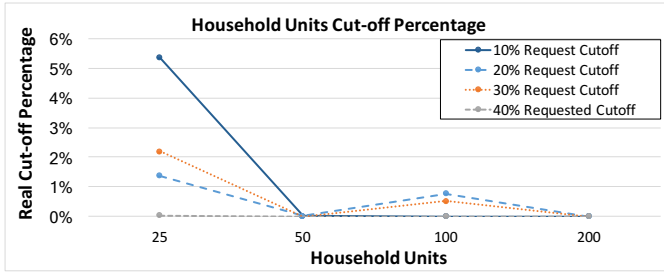


Fig. 9. Cut-off percentage exceeding nominal request.

Increasing the number of appliances leads to more tight results. Power profiles of the different appliances have been generated so that: i) the duration is indicated in minutes and is uniformly randomly generated in the range [80...160]; ii) the energy consumption for each minute is uniformly randomly generated in the range [0.5...1.5] kW. The cut-off requests are 10%, 20%, 30% and 40% of the global consumption generated by the whole set of appliances.

Fig. 9 shows the relation between the number of *Household Units* of a specific cut-off request and the increase in the cut-off quantity calculated by the algorithm with respect to the nominal "optimal" cut-off obtained by the simple percentage calculation. Clearly the nominal cut-off can also partially deactivate an appliance, thing which is not possible in reality since an appliance is switched-on or off. As can be seen, if more *Household Units* are considered, the algorithm tends to perform better since there is more room for optimization than with fewer appliances. In general, the worst case is reached for 25 *Household Units* only and never exceeds 5.5%. This is caused by the higher impact of a single appliance w.r.t. the global consumption. Besides, notice that this is only a rough estimation of the percentage error as the discreteness of appliances activation/deactivation is not taken into account at all in this estimation.

### C. Scalability performance of the framework

In this section, we present results about the scalability of the proposed co-simulation framework. We performed stress-tests with 1k, 5k and 10k *Household Units*, respectively, to evaluate the computation performance of i) the FLEXMETER infrastructure, ii) the *Energy Aggregation Platform* and iii) the proposed DR heuristic running in EAP. On average, we considered five appliances per *Household Units* that can be managed. We also present results on the performance of data transmission among the different actors in our framework.

As mentioned in Section III-D, the various components of the proposed framework can be installed in different locations and they communicate over the Internet. Hence, to perform these stress-tests, a single instance of EAP was deployed in a physical server in our data center with a CPU Intel Xeon 3.40 GHz with 8 cores and 32 GB of RAM. Whilst, the FLEXMETER platform was running in a virtual server in a cloud with a CPU Intel Xeon 2.20 GHz with 16 cores and 24 GB of RAM. To perform these tests, we developed a special TIA to simulate the communication behaviour of smart

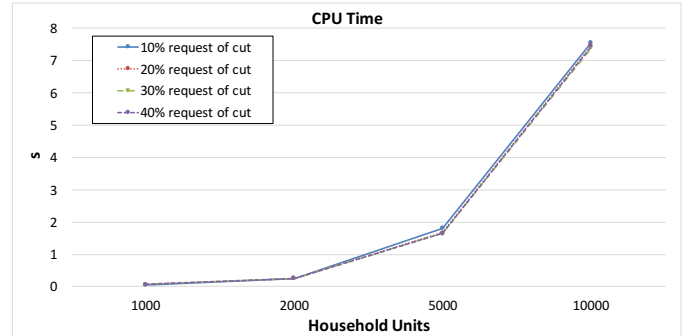


Fig. 10. CPU time trend w.r.t. Household Units.

appliances. EAP, FLEXMETER and TIA have been deployed in Italy, Romania and Germany, respectively. According to our tests, increasing the number of *Household Units* from 1k to 10k, thus the number of smart appliances, the usage of CPU and RAM was almost constant for both EAP and FLEXMETER. EAP had less than 5% CPU and RAM occupancy. Whilst, FLEXMETER occupied CPU less than 30% and RAM between 70% and 80%.

Fig. 10 benchmarks the CPU time in seconds needed by the DR-algorithm to perform optimization for the different classes of requested cut-offs. Results confirm that the trend of the CPU time is basically linear with respect to the number of *Household Units*. The increase in the CPU time is consistent with the increase in the number of appliances of the problem instance set. This can also assess the upper limits in term of problem sizes if the utility needs to set up a maximum time for computing a solution of the DR. Also in these tests, the cut-off requests are 10%, 20%, 30% and 40% of the global consumption generated by the whole set of appliances. As shown in Fig. 10, the computational time needed by the DR algorithm is almost the same for the different cut-off requests and it varies from about 0.06s to about 7.5s for 1k and 10k *Household Units*, respectively.

To assess the transmission performance of the proposed infrastructure over the Internet, we performed communication tests to estimate latency in retrieving data and sending actuation commands. The results of these tests are shown in the box-plots of Fig. 11.

Fig. 11-(a) shows the latency values when sending measures from the smart devices to FLEXMETER infrastructure over MQTT protocol, with a payload of about 150 byte. As shown by the graph, data transmission is not affected by the variation of *Household Units*, maintaining a median value of about 70ms for *Household Units* varying from 1k to 10k. Furthermore, the variability of the latency values is in general very low.

Fig. 11-(b) shows the latency values of REST Web Service when retrieving information from FLEXMETER. More specifically, EAP invokes a specific REST Web Service to obtain information from all the *Household Units* involved in the DR-event. As shown in Fig. 11-(b), the latency is generally very low (less than about 8s).

Fig. 11-(c) reports the latency of actuation commands sent from EAP to smart devices through the FLEXMETER infras-



Fig. 11. Transmission latency at different Household Units: (a) MQTT, (b) REST, (c) Commands.

structure. This latency includes: i) the time needed by EAP to post the actuation commands to FLEXMETER via REST and ii) the time needed by FLEXMETER to send such commands to the smart devices via MQTT. Each actuation command is intended with a payload of about 300 byte. Even in this case, the graph shows no significant variation of latency at different *Household Units* (about 150ms).

It is worth noting that we performed these stress-tests to evaluate the correct behaviour of the overall framework. In a realistic scenario and following the de-centralization approach of Smart Grids management, different instances of both EAP and FLEXMETER can be deployed in cloud systems or data centers geographically closed to minimize the delay on data transmission. We can also suppose that different EAP and FLEXMETER instances can manage different portion of the Smart Grid or different Microgrids.

## VI. CONCLUSION

In this paper, we presented a novel distributed framework for both (near-) real-time management and co-simulation of DR-policies in Smart Grids. We also discussed how the components of the proposed framework can be deployed in a distributed environment compliant with the *Internet-of-Energy* view. The usability of our solution was demonstrated on a realistic smart grid, where a test case DR-policy was applied. In this scenario, we discussed and compared the results of total apparent power at MV/LV substation and the voltage profiles at different nodes with and without DR. To perform these tests, we deployed components of our framework in Italy, Romania and Germany discussing the advantages of having a distributed co-simulation platform and presenting its performance.

In our view, limitations and challenges to deploy the proposed solution in real-world scenarios are related to a lack of current regulatory frameworks instead of technological issues. Moreover, the current marketplace does not include real incentives for customers that will help to spread DR solutions. Another limitation consists on rising smart appliances that are not yet widespread at customer's home premises, but they will be in the future years.

## REFERENCES

- [1] P. Capros, A. D. Vita, N. Tasios, D. Papadopoulos, P. Siskos, E. Apostolaki, M. Zampara, L. Paroussos, K. Fragiadakis, N. Kouvaritakis, L. Hoglund-Isaksson, W. Winiwarter, P. Purohit, H. Bottcher, S. Frank, P. Havlik, M. Gusti, and H. P. Witzke, *EU energy, transport and ghg emissions trends to 2050*. Wiley, 2013.
- [2] A. Mohd, E. Ortjohann, A. Schmelter, N. Hamsic, and D. Morton, "Challenges in integrating distributed energy storage systems into future smart grid," in *Industrial Electronics, 2008. ISIE 2008. IEEE International Symposium on*. IEEE, 2008, pp. 1627–1632.
- [3] J. S. Vardakas, N. Zorba, and C. V. Verikoukis, "A survey on demand response programs in smart grids: Pricing methods and optimization algorithms," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 1, pp. 152–178, 2015.
- [4] R. Deng, Z. Yang, M.-Y. Chow, and J. Chen, "A survey on demand response in smart grids: Mathematical models and approaches," *IEEE Transactions on Industrial Informatics*, vol. 11, no. 3, pp. 570–582, 2015.
- [5] P. T. Baboli, M. P. Moghaddam, and M. Eghbal, "Present status and future trends in enabling demand response programs," in *Power and Energy Society General Meeting, 2011 IEEE*. IEEE, 2011, pp. 1–6.
- [6] T.-C. Chiu, Y.-Y. Shih, A.-C. Pang, and C.-W. Pai, "Optimized day-ahead pricing with renewable energy demand-side management for smart grids," *IEEE Internet of Things Journal*, vol. 4, no. 2, pp. 374–383, 2017.
- [7] B. P. Bhattarai, B. Bak-Jensen, P. Mahat, J. R. Pillai, and M. Maier, "Hierarchical control architecture for demand response in smart grids," in *Power and Energy Engineering Conference (APPEEC), 2013 IEEE PES Asia-Pacific*. IEEE, 2013, pp. 1–6.
- [8] C.-C. Lin, D.-J. Deng, W.-Y. Liu, and L. Chen, "Peak load shifting in the internet of energy with energy trading among end-users," *IEEE Access*, vol. 5, pp. 1967–1976, 2017.
- [9] Y. Wang, S. Mao, and R. M. Nelms, "Distributed online algorithm for optimal real-time energy distribution in the smart grid," *IEEE Internet of Things Journal*, vol. 1, no. 1, pp. 70–80, 2014.
- [10] F. Y. Xu and L. L. Lai, "Novel active time-based demand response for industrial consumers in smart grid," *IEEE Transactions on Industrial Informatics*, vol. 11, no. 6, pp. 1564–1573, 2015.
- [11] US Department of Energy, "Benefits of demand response in electricity markets and recommendations for achieving them," *US Department of Energy, Washington, DC, USA, Tech. Rep*, 2006.
- [12] P. Bertoldi, P. Zancanella, and B. Boza-Kiss, "Demand response status in eu member states," *Europa. eu: Brussels, Belgium*, 2016.
- [13] European Commission, "Directive 2012/27/eu of the european parliament and of the council of 25 october 2012 on energy efficiency, amending directives 2009/125/ec and 2010/30/eu and repealing directives 2004/8/ec and 2006/32," *Official Journal, L*, vol. 315, pp. 1–56, 2012.
- [14] M. Yun and B. Yuxin, "Research on the architecture and key technology of internet of things (iot) applied on smart grid," in *Advances in Energy Engineering (ICAEE), 2010 International Conference on*. IEEE, 2010, pp. 69–72.
- [15] M. Rana, "Architecture of the internet of energy network: An application to smart grid communications," *IEEE Access*, vol. 5, pp. 4704–4710, 2017.

- [16] D. Niyato, L. Xiao, and P. Wang, "Machine-to-machine communications for home energy management system in smart grid," *IEEE Communications Magazine*, vol. 49, no. 4, 2011.
- [17] J. Lin, W. Yu, N. Zhang, X. Yang, H. Zhang, and W. Zhao, "A survey on internet of things: Architecture, enabling technologies, security and privacy, and applications," *IEEE Internet of Things Journal*, vol. 4, no. 5, pp. 1125–1142, 2017.
- [18] M. Chiang and T. Zhang, "Fog and iot: An overview of research opportunities," *IEEE Internet of Things Journal*, vol. 3, no. 6, pp. 854–864, 2016.
- [19] N. Bui, A. P. Castellani, P. Casari, and M. Zorzi, "The internet of energy: a web-enabled smart grid system," *IEEE Network*, vol. 26, no. 4, 2012.
- [20] K. Boroojeni, M. H. Amini, A. Nejadpak, T. Dragičević, S. S. Iyengar, and F. Blaabjerg, "A novel cloud-based platform for implementation of oblivious power routing for clusters of microgrids," *IEEE Access*, vol. 5, pp. 607–619, 2017.
- [21] H. Tram, "Advanced metering infrastructure as an enabler of demand response," in *Transmission and Distribution Conference and Exposition, 2010 IEEE PES*. IEEE, 2010, pp. 1–1.
- [22] Q. Sun, H. Li, Z. Ma, C. Wang, J. Campillo, Q. Zhang, F. Wallin, and J. Guo, "A comprehensive review of smart energy meters in intelligent energy networks," *IEEE Internet of Things Journal*, vol. 3, no. 4, pp. 464–479, 2016.
- [23] E. Patti, A. L. A. Syri, M. Jahn, P. Mancarella, A. Acquaviva, and E. Macii, "Distributed software infrastructure for general purpose services in smart grid," *IEEE Transactions on Smart Grid*, vol. 7, no. 2, pp. 1156–1163, 2016.
- [24] A. Bahmanyar, S. Jamali, A. Estebarsari, E. Pons, E. Bompard, E. Patti, and A. Acquaviva, "Emerging smart meters in electrical distribution systems: Opportunities and challenges," in *Electrical Engineering (ICEE), 2016 24th Iranian Conference on*. IEEE, 2016, pp. 1082–1087.
- [25] M. A. Mengistu, A. A. Girmay, C. Camarda, A. Acquaviva, and E. Patti, "A cloud-based on-line disaggregation algorithm for home appliance loads," *IEEE Transactions on Smart Grid*, 2018.
- [26] N. G. Paterakis, O. Erdinc, A. G. Bakirtzis, and J. P. Catalão, "Optimal household appliances scheduling under day-ahead pricing and load-shaping demand response strategies," *IEEE Transactions on Industrial Informatics*, vol. 11, no. 6, pp. 1509–1519, 2015.
- [27] E. Bompard, A. Monti, A. Tenconi, A. Estebarsari, T. Huang, E. Pons, M. Stevic, S. Vaschetto, and S. Vogel, "A multi-site real-time co-simulation platform for the testing of control strategies of distributed storage and v2g in distribution networks," in *2016 18th European Conference on Power Electronics and Applications (EPE'16 ECCE Europe)*, Sept 2016, pp. 1–9.
- [28] M. Stevic, A. Estebarsari, S. Vogel, E. Pons, E. Bompard, M. Masera, and A. Monti, "A multi-site european framework for real-time co-simulation of power systems," *IET Generation, Transmission & Distribution*, 2017.
- [29] M. Stevic, S. Vogel, M. Grigull, A. Monti, A. Estebarsari, E. Pons, T. Huang, and E. Bompard, "Virtual integration of laboratories over long distance for real-time co-simulation of power systems," in *Industrial Electronics Society, IECON 2016-42nd Annual Conference of the IEEE*. IEEE, 2016, pp. 6717–6721.
- [30] F. Aguirre, A. Silvestris, J. Alemany, and F. Magnago, "A low cost smart metering infrastructure with centralized demand response management," in *Innovative Smart Grid Technologies Latin America (ISGT LATAM), 2015 IEEE PES*. IEEE, 2015, pp. 520–524.
- [31] M. LeMay, R. Nelli, G. Gross, and C. A. Gunter, "An integrated architecture for demand response communications and control," in *Hawaii international conference on system sciences, proceedings of the 41st annual*. IEEE, 2008, pp. 174–174.
- [32] D. Mashima and W.-P. Chen, "Residential demand response system framework leveraging iot devices," in *Smart Grid Communications (SmartGridComm), 2016 IEEE International Conference on*. IEEE, 2016, pp. 514–520.
- [33] H. Kim, Y.-J. Kim, K. Yang, and M. Thottan, "Cloud-based demand response for smart grid: Architecture and distributed algorithms," in *Smart Grid Communications (SmartGridComm), 2011 IEEE International Conference on*. IEEE, 2011, pp. 398–403.
- [34] R. H. Jacobsen, A. G. Azar, and E. S. M. Ebeid, "Design of an event-driven residential demand response infrastructure," in *Digital System Design (DSD), 2016 Euromicro Conference on*. IEEE, 2016, pp. 38–45.
- [35] B. Bhattarai, M. Levesque, B. Bak-Jensen, J. Pillai, M. Maier, D. Tipper, and K. Myers, "Design and co-simulation of hierarchical architecture for demand response control and coordination," *IEEE Transactions on Industrial Informatics*, 2016.
- [36] S. C. Müller, H. Georg, J. J. Nutaro, E. Widl, Y. Deng, P. Palensky, M. U. Awais, M. Chenine, M. Kuch, M. Stifter *et al.*, "Interfacing power system and ict simulators: Challenges, state-of-the-art, and case studies," *IEEE Transactions on Smart Grid*, 2016.
- [37] C.-h. Yang, G. Zhabelova, C.-W. Yang, and V. Vyatkin, "Cosimulation environment for event-driven distributed controls of smart grid," *IEEE Trans. on Industrial Informatics*, vol. 9, no. 3, pp. 1423–1435, 2013.
- [38] M. Manbachi, A. Sadu, H. Farhangi, A. Monti, A. Palizban, F. Ponci, and S. Arzanpour, "Real-time co-simulation platform for smart grid volt-var optimization using iec 61850," *IEEE Trans. on Industrial Informatics*, vol. 12, no. 4, pp. 1392–1402, Aug 2016.
- [39] S. Schütte, S. Scherfke, and M. Tröschel, "Mosaik: A framework for modular simulation of active components in smart grids," in *Proc. of SGMS11*. IEEE, 2011, pp. 55–60.
- [40] S. Rohjans, S. Lehnhoff, S. Schtte, S. Scherfke, and S. Hussain, "mosaik - a modular platform for the evaluation of agent-based smart grid control," in *Proc. of PES ISGT*, Oct 2013, pp. 1–5.
- [41] T. Hess, J. Dickert, and P. Schegner, "Multivariate power flow analyses for smart grid applications utilizing mosaik," in *Proc. of PES ISGT*, Oct 2016, pp. 1–6.
- [42] K. Anderson, J. Du, A. Narayan, and A. El Gamal, "Gridspice: A distributed simulation platform for the smart grid," *IEEE Trans. on Industrial Informatics*, vol. 10, no. 4, pp. 2354–2363, 2014.
- [43] M. Brenna, E. De Berardinis, L. D. Carpinì, F. Foadelli, P. Paulon, P. Petroni, G. Sapienza, G. Scrosati, and D. Zaninelli, "Automatic distributed voltage control algorithm in smart grids applications," *IEEE Trans. on Smart Grid*, vol. 4, no. 2, pp. 877–885, 2013.
- [44] P. Faria, Z. A. Vale, and J. Ferreira, "Demisia demand response simulator in the context of intensive use of distributed generation," in *Systems Man and Cybernetics (SMC), 2010 IEEE International Conference on*. IEEE, 2010, pp. 2025–2032.
- [45] S. Tan, W.-Z. Song, Q. Dong, and L. Tong, "Score: Smart-grid common open research emulator," in *Smart Grid Communications (SmartGridComm), 2012 IEEE Third International Conference on*. IEEE, 2012, pp. 282–287.
- [46] T. K. Wijaya, D. Banerjee, T. Ganu, D. Chakraborty, S. Battacharya, T. Papaioannou, D. P. Seetharam, and K. Aberer, "Drsim: A cyber physical simulator for demand response systems," in *Smart Grid Communications (SmartGridComm), 2013 IEEE International Conference on*. IEEE, 2013, pp. 217–222.
- [47] M. Pau, E. Patti, L. Barbierato, A. Estebarsari, E. Pons, F. Ponci, and A. Monti, "A cloud-based smart metering infrastructure for distribution grid services and automation," *Sustainable Energy, Grids and Networks*, 2017.
- [48] M. Pathirage, S. Perera, I. Kumara, and S. Weerawarana, "A multi-tenant architecture for business process executions," in *Web services (icws), 2011 IEEE international conference on*. IEEE, 2011, pp. 121–128.
- [49] E. Patti, A. Acquaviva, and E. Macii, "Enable sensor networks interoperability in smart public spaces through a service oriented approach," in *Advances in Sensors and Interfaces (IWASI), 2013 5th IEEE International Workshop on*, June 2013, pp. 2–7.
- [50] "Message Queue Telemetry Transport (MQTT)," Available: <http://mqtt.org/>.
- [51] P. T. Eugster, P. A. Felber, R. Guerraoui, and A.-M. Kermarrec, "The many faces of publish/subscribe," *ACM CSUR*, June 2003.
- [52] R. T. Fielding and R. N. Taylor, "Principled design of the modern web architecture," *ACM Transactions on Internet Technology*, vol. 2, no. 2, pp. 115–150, 2002.
- [53] A. Aliberti, C. Camarda, V. Ferro, A. Acquaviva, and E. Patti, "A participatory design approach for energy-aware mobile app for smart home monitoring," pp. 158–165, 2017.
- [54] L. Bottaccioli, A. Estebarsari, E. Pons, E. Bompard, E. Macii, E. Patti, and A. Acquaviva, "A flexible distributed infrastructure for real-time co-simulations in smart grids," *IEEE Transactions on Industrial Informatics*, 2017.
- [55] T. H. Cormen, *Introduction to algorithms*. MIT press, 2009.
- [56] S. Martello and P. Toth, *Knapsack Problems: Algorithms and Computer Implementations*. John Wiley & Sons, Inc., 1990.